VOL. 5 NO. 7
JULY 1983

neat notes

THE NEWSLETTER OF NEW ENGLAND APPLE TREE

July 1983 N.E.A.T. Meeting

The July meeting will be held at 7:30 P.M. on the third Wednesday of the month (as usual), July 20, 1983 at the Mitre cafeteria. Mitre is on rte 62 in Bedford Ma. Take route 3 north from rte 128 to the first exit (rte 62).

JULY'S MEETING

Our speaker for July is Joe Stivaletts, our own membership chairman. Joe will present the new ARCADE BOARD for the Apple, a special board which gives you animated graphics and sound generation. This board uses many of the same chips found in the coin-operated arcades, and can produce arcade-style graphics from BASIC. See the accompanying article on Page 5.

AN EXCITING SCHEDULE IS SHAPING UP FOR THE SUMMER!
THE FUTURE SCHEDULE IS TENTATIVELY AS FOLLOWS:

July 20,1983----------ARCADE BOARD, Joe Stivaletta

August 17,1983--------68000 in an Apple,
Joe Stivaletta

September 21,1983-----CompuServ

October 19,1983-------Lisa, Jim Miller

November 16,1983------

December 21,1983------

ANA-LIST™

THE PROCESSOR FOR LISTS AND TABLES

IF MANY OF YOUR DATA BASES ARE REALLY LISTS

TRY ANA-LIST

YOU'LL SAVE TIME AND AGGRAVATION — ANA-LIST SORTS IN SECONDS — YOU PREPARE CUSTOM REPORT FORMATS IN LESS THAN TWO MINUTES

IF MANY OF YOUR SPREADSHEETS ARE REALLY LISTS

TRY ANA-LIST

YOU'LL SAVE TIME AND AGGRAVATION — ANA-LIST DOES THE SORTING FOR YOU IN SECONDS — YOU CAN REPORT THE COLUMNS AND ROWS YOU WANT IN THE FORMAT YOU DEFINE

IF YOU CURRENTLY MAKE LISTS WITH PAPER AND PENCIL

TRY ANA-LIST

YOU'LL SAVE TIME AND EASILY REPORT JUST THE DATA APPROPRIATE FOR EACH PURPOSE

WITH ANA-LIST YOU CAN:

- **SORT IN SECONDS**
- **PRODUCE CUSTOMIZED REPORTS**
- **DO CALCULATIONS AUTOMATICALLY**
- **ENTER AND CHANGE DATA QUICKLY**

TYPICAL APPLICATIONS

BUSINESS	HOME	PROFESSIONAL
SALES PROSPECTS	HOUSEHOLD INVENTORY	DAILY & WEEKLY SCHEDULES
VENDOR LISTS	VIDEO CASSETTE LIBRARY	TICKLER LISTS
COMPETITIVE BIDS	INSURANCE COVERAGE	CLIENT PROSPECTS
SHIPMENT SCHEDULES	INVESTMENT PORTFOLIO	DAILY & WEEKLY BILLINGS
MATERIAL SHORTAGES	IMPORTANT DATES	CLASS LISTS
SALARY REVIEWS	TELEPHONE LIST	GRADE BOOKS
ACCOUNT ANALYSIS	HOME LIBRARY CATALOG	PERSONAL LIBRARY

ANA-LIST IS QUICK, EASY TO LEARN & SIMPLE TO USE • MANAGES LISTS FROM 1 TO 46 COLUMNS • 48K: TYPICAL LIST OF 6 COLUMNS HAS 100 ROWS • 64K: TYPICAL LIST OF 6 COLUMNS HAS 400 ROWS • SORTS ON ONE KEY COLUMN AND UP TO 9 SUBSORTS • DATA AND PROGRAM ARE MEMORY RESIDENT • ANA-LIST PACKAGE INCLUDES 150 PAGE MANUAL WITH TUTORIAL AND REFERENCE SECTIONS, TWO COPIES OF ANA-LIST PROGRAM DISK, AND SAMPLE STORAGE DISK WITH EXAMPLES • SYSTEM REQUIREMENTS: APPLE II+ OR APPLE IIe, 48K OR 64K; 1 OR 2 DISK DRIVES

PRICE: $150 plus $3.50 for shipping and handling — VISA AND MASTERCARD ACCEPTED

TO ORDER CALL OR WRITE:

SYNOPTIC SOFTWARE, INC. 57 RESERVOIR LANE, CHESTNUT HILL, MA 02167

(617) 277-0778

neat notes

THE NEWSLETTER OF NEW ENGLAND APPLE TREE

NEAT NOTES STAFF

Editor/Publisher.....
John Shaw,643-2679
Editor/Basic.....
Jack Sheehan,924-1197
Editor/Writer-Assembly Language
John Shaw,643-2679
Games Reviewer.....
Mark Hessman,475-2617
Advertising Manager....
HELP !!!

N.E.A.T. OFFICERS and CHAIRMEN

President..Terry Scheller,244-4833
Vice-pres...Jack Thompson,655-1096
Vice-pres....Elliot Entin,862-6418
Secretary....Alice Janian,623-6757
Treasurer.....David Lynn,655-6164
Library.....
Input.....M.Gooding,264-4237
Sales.....D.McMullin,444-4304
DOM (at D.Goldstein,273-0567
meeting).E.Mac Lauchlan,878-2229
Outside User Group Contact.....
Larry Starnes,263-1795
Membership.Joe Stivaletta,861-1899
Auditor.....Don Foster
Greeter.....Jack Thompson,253-8608

SPECIAL INTEREST GROUPS

SIGED.....Jack Sheehan,924-1197
SIGCP/M....Joe Stivaletta,861-1899
SIGASSY.....John Shaw,643-2679
SIGCOMM.....
SIGPASCAL..Jerry Connelly,497-9050
SIGNEW.....Lou Tarsa,547-3289

Copyright 1982, New England Apple Tree, All rights reserved.

All original Articles may be reprinted without permission by computer club newsletters if they are not copyrighted by the author. Reprints must clearly indicate the author's name, the name and address of NEAT, a copyright notice and the date of the issue. Two copies of the document in which the reprint appears should also be sent to:
Attn. Editor,NEAT Notes
P.O.Box 2652
Woburn,Ma 01801

JULY HIGHLIGHTS

NEAT Meeting Notes
by Alice Janian.....Page 4
Calendar
by Raines Cohen.....Page 4
JULY SPEAKER: Joe Stivaletta
.....Page 5
Boston Applefest: A Review
.....Page 6
A New Computer Store
.....Page 7
Local Computer Bulletin Boards
Bette Winer.....Page 7
Copy Protect 'Your' Disks
Martin Piercey.....Page 9
A New DOS?
The DOS Mouse.....Page 10
Review: TG Trackball
The Winers.....Page 11
Just the Basics
Jack Sheehan.....Page 12
Product Announcement: ANALIST
.....Page 13
Macrosoft: A Review
Dr. J.....Page 14
Machine Language Spoken Here
Dr. J.....Page 15
Pascal Disk Notes
Jim Wrenholt.....Page 18
Rear Guard: A Review
Hugh Kurtzman.....Page 21
Membership Application and
DOM Order Blank.....Page 23

N.E.A.T. MEETING MINUTES
June 15, 1983

Present: Lou Tarsa, Dave Reed, Joe Stivaletta, John Shaw.

Lou Tarsa called the meeting to order at 8 P.M. and introduced new members to the group) It was a hot night (no air conditioning).

New Business - Election of new officers: Dave Reed read the names of the new officers that volunteer to serve:

President.....Terry Scheller
Vice President..Jack Thompson
Vice President...Elliot Entin
Treasurer.....Dave Lynn
Secretary.....Alice Janian
Editor/Publisher....John Shaw
Membership.....Joe Stivaletta
Auditor.....Don Foster

Since there was no competition-everybody was voted in.

The Disk of the Month was "Machine Language Disk", which included a number of utilities that aid in writing and debugging machine language programs. The disk also included a Machine Language Tutorial.

Our new President, Terry Scheller, asked all the members for ideas and projects. Members were asked to write them down on a sheet of paper and pass it in after the meeting.

This Month's Guest Speaker was David P. Szetela - Technical Editor of Nibble Magazine. The subject was "How to Publish Your New Software". We also had a Question and Answer Period. The discussion was very enlightening.

Meeting adjourned at 9:15 P.M.

CALENDAR
Compiled by Raines Cohen

July 27 7:30PM
Boston Computer Society
New England Life Hall
call 227-9178

July 28
BCS Pascal User's Group
MIT 1-146
call 227-9178

July 30 2:30PM
Newton Apple Kids
Raines Cohen's House
call 965-0033

August 3 7:00PM
New England Computer Society
Mitre Cafeteria
call 646-3080

August 13 10:00AM
North Shore Apple Group
Computer City, Danvers
call 546-3104

August 17 7:30PM
New England Apple Tree
Mitre Cafeteria
call 497-9050

August 20 9:30AM
Apple Guild
Massasoit Community College
call 767-1303

August 27 2:30PM
Newton Apple Kids
Raines Cohen's House
call 965-0033

JULY SPEAKER: JOE STIVALETTA

Due to a last minute engagement, Jim Miller will not be able to demonstrate the new Lisa computer at the July NEAT meeting. The good news is our intrepid Membership Chairman, Joe Stivaletta, has an exciting talk about the new ARCADE BOARD lined up for us. The ARCADE BOARD, by Third Millennium Engineering Corp., uses the same hardware found in many of the coin-operated arcade machines to give you true arcade quality graphics, animation, music, and sound effects on your Apple.

Why would you want such a board? Well, on a standard Apple all instructions for drawing pictures and creating sounds must go through the Apple's lonely 6502 processor. Advanced as the 6502 is, it just isn't fast enough to offer smooth animation and intricate sound effects at the same time. (You may notice, for instance, that many arcade-style games for the Apple do offer music and sound effects, but these sounds are normally played when there is no motion on the screen.) The arcade manufactures would face this same problem if they did not design special graphics and sound generating chips to take some of the workload off the central processor. For example, while the central processor is busy calculating new positions for objects on the screen, a second chip handles the actual drawing of the pictures, and a third chip handles all sound generation. Very advanced games, such as BATTLE ZONE with its three-dimensional graphics, use three or more processors to bring their amazing effects to the screen. Third Millinneum's ARCADE BOARD combines these same chips onto a single board for the Apple where they may be controlled from BASIC (and assembly language). The upshot of all this is you or I can now sit down, write a game in Applesoft BASIC (a language most of us know) and get **better** graphics and sound effects than we could were we to program a standard Apple using assembly language (a language many of us do not know)!

The ARCADE BOARD uses a special video display processor and a programmable sound generator. The board can support up to 32 SPRITES (programmable hi-resolution symbols that can be used for animation), will allow 16 colors simultaneously in the high-resolution graphics mode (normal hi-res graphics allows only six colors), and has 16K of RAM which can store up to 14 pages of hi-res graphics. The sound generator has three tone generators, a nine octave range, and 16 level volume control (you can now design games to be played late at night!). The sound generator can also simulate a number of musical instruments, including a piano, drums, chimes, etc.

The suggested list price is $295, although club members can purchase it for $165. You can receive more information about this board by writing to Third Millinneum Engineering Corp at:

1015 Gayley Ave.
Suite 394
Los Angeles, CA 90024

BOSTON APPLEFEST: A REVIEW

Last month, we reviewed several hardware products which were shown at Boston's APPLEFEST in May. We complete the review this month with a discussion of some of the software which was introduced at this show.

First, several BIG West Coast software firms were absent this year, including Broderbund, Datamost, VisiCorp, Sierra On-Line, and Sirius. We failed to see, therefore, the introduction of major games for the Apple this year. Nevertheless, several large firms (e.g., Penguin, Sir-tech, BPI) were present along with many new (and hopefully up and coming) software firms to show us several new business, graphics, communication, and utility packages.

MicroSPARC brought several new products to the show. For all you assembly language hackers, MicroSPARC introduced two new products: THE ASSEMBLER, a macro-assembler and editor for the Apple][/][+/][e, and MACROSOFT, a special library for THE ASSEMBLER (and only THE ASSEMBLER) which lets you write Applesoft-like programs directly in machine language [see the review of this product in this issue]. Looking for telecommunications software? MicroSPARC also brought out TELETEXT, which turns your Apple into a 'smart' terminal. With TELETEXT, you can send and receive BASIC, text, binary, FORTRAN, and Pascal files to and from your Apple and another computer. TELETEXT is compatible with the 80 column Videx card, the Hayes Micromodem II, the Novation Apple Cat II, and any modem for the Apple Communications Card.

The COMPUTER MART in Waltham brought out a version of their TechWriter word processing software for the Apple][. This package is the only word processing software to support scientific and engineering equations and symbols within a document. Furthermore, TechWriter does this all on the Apple's High Resolution screen, and does not require an 80-column card. TechWriter's screen-oriented editor shows your document on your screen just as it will appear when it is printed; no more guessing whether the margins and underlining are correct!

PRIMESOFT in Maryland introduced their new PRIME PLOTTER graphics software at the show. I was very impressed with this package: it can do scatter, line, bar, and pie charts in 2-dimensional and 3-dimensional plots! For example, the pie charts you can create are quite remarkable: 3-D (volume and area) color filling with exploded slices; I've seen 'professional' graphics programs for main-frame computers (and selling for mucho dollars) that could not match these capabilities. PRIME PLOTTER also offers several useful statistics routines, including regression (linear, logarithmic, exponential, inverse log), trend analysis (averaging, moving average, cumulative histograms), and frequency distribution (gamma, normal, Chi-square). Finally, PRIME PLOTTER also supports Software Art's DIF file format, meaning you can create

graphs from VISICALC (and compatible) files.

Exec Software (in Waltham, Mass.) introduced a new telecommunications package for the Apple: TERMEXEC. I did not see this software in action, but the literature claims TERMEXEC offers backscrolling (to recall lines which have scrolled off the top of the screen), lateral scrolling (to display 80 columns on your normal 40 column screen), and unattended file capture which receives files and saves them directly on your disk. TERMEXEC also apparently offers a full screen editor, and 'macro' facilities to permit instant execution of long command sequences. All told, TERMEXEC appears to have several powerful features and should be worth checking if you are looking for telecommunications software for your machine. You can call Exec Software at (617) 862-3170.

For all you pfs:GRAPH fans, National Software (in Dover, Mass.) introduced a handy utility which allows you to save your Apple][pfs:GRAPH pictures in a standard DOS (binary) file. When pfs:GRAPH saves a picture, it simply saves the data values and picture definition parameters, not the picture itself. GRAPH:SAVE can be used in conjunction with pfs:GRAPH to save the actual picture in a binary file. I bought this utility and have been using it at work with great success. At $19, GRAPH:SAVE is a nice bargain. The address for National Software is:

P.O. BOX 686
Dover, MA 02030

That wraps up our APPLEFEST review for this year. See you again in '84!

A NEW COMPUTER STORE FOR PEABODY

This past July fourth weekend, a new computer store "Computer's etc." opened in Peabody, Mass. on the southbound side of Rte 1, next to the Plywood Ranch. I talked to the folks at Computers etc. and asked them what services they will offer to computer users. Owner, Howard Gelpey, replied: "As an Apple owner and a computer hobbyist for years, I felt a lack of support and service whenever I visited various computer stores in the New England area. Most stores are more than happy to sell you a system, but never seem to offer the same kind of attention when you want to purchase a small item or just want information on interfacing or software, etc.

My staff and I hope to provide service and support to who ever walks in for whatever reason. I hope that all NEAT members will stop in to see our new store and say hi to the staff. We're open every day of the week (M-F 9:30-9, Sat 9-5, Sun 12-5). Through your support we hope to support you."

LOCAL COMPUTER BULLETIN BOARD SYSTEMS

The following is a list of all active boards in the 617 (E. MASS) AND 603 (S. NH) area codes as published on the Boston Bullet Computer Bulletin Board. If you have any additions, corrections, deletions or questions, please contact the Boston Bullet.

NAME OF BOARD	LOCATION	PHONE #	HOURS	SYSOP
********** 617 AREA CODE - 27 ACTIVE BOARDS **********				
		BINEX FAMILY		
BOSTON-BULLET	BOSTON	266-7789	24 HRS	M GOLDBERG
SPEED BOARD	BOSTON	254-4327	24 HRS	S PHREAK
N. SHORE AMIS	LYNN	595-0211	24 HRS	B OLIVA
OUTPOST TBBS	LINCOLN	259-0181	24 HRS	R SHWARTZ
MACRO EXCHANGE	BILLERICA	667-7388	24 HRS	J HAJINLIAN
BINEX HUB & RCP/M	BOSTON	423-6985	24 HRS	G MEADER
		OTHER BOARDS		
T-I-M-C-O-R	BOSTON	720-3600	24 HRS	R REBEL
NECS CBBS	ARLINGTON	646-3610	24 HRS	D MITTON
RCP/M	LEXINGTON	862-0781	1900-0900	P KELLY
COMP BARGAIN GUIDE	WALTHAM	899-6524	24 HRS	J WALLACE
SBSG, INC.	WESTFORD	692-3973		B DRISCOLL
MILFORD MASS RC/PM	MILFORD	478-4164	24 HRS	H MOULTON
LAWRENCE GEN HOSP CBBS	LAWRENCE	683-2119	24 HRS	M CUSHING
N SHORE HEATH UG BBS	PEABODY	531-9332	24 HRS	D RUBIN
MEMORY ALPHA	N BEDFRD	999-3210		B PERRY
DIAL-YOUR-MATCH-18	LYNFIELD	334-6369		MATCHMAKER
ANDOVER CNODE	ANDOVER	470-2548	24 HRS	L DUBOSE
MICROSTAR CBB	WORCESTER	752-7284	1200-0800	R MCFATTER
WELLSLY HEATH UG BBS	WELLESLY	237-1511	1830-0900	D GENTILE
BCS BOARD	NEWTON	969-9660	24 HRS	K PUGH
COMP AD. NET	BOSTON	423-6300	24 HRS	CAN
ALPHA CONTROL	CHELMSFORD	937-1970	24 HRS	G FRANCIS
		PIRATE'S NET BOARDS:		
PIRATE'S CHEST	LINCOLN	891-1349	24 HRS	J KOUYOUMJIAN
PIRATE'S HIDEAWAY	NEEDHAM	449-2808	$ 24 HRS	T FALCON
PIRATE'S PUB	WESTON	894-7266	$ 24 HRS	T BARTENDER
PIRATE'S PUB II	LINCOLN	891-5793	$ 24 HRS	T BOUNCER
TREASURE ISLAND	NEEDHAM	444-6821	24 HRS	T MARAUDER
********** 603 AREA CODE - 4 ACTIVE BOARDS **********				
NASHUA NH BBS	NASHUA	888-4488	24 HRS	S PETERFREUND
ACCESS-80	NASHUA	888-6999	24 HRS	C OROPALLO
MAGAZINE-80	PETERBRO	924-7920	24 HRS	B LIDDIL
SOFTWARE REFERRAL SERVICE	MANCHSTR	625-1919	0900-2330	P JOHNSON

* = 300/1200 BAUD SUPPORTED $ = FEE REQUIRED

COPY PROTECT 'YOUR' DISKS

by Martin Piercey

(Reprinted from: the Maple Orchard, Willowdale, Ontario.)

Residing within DOS is the 'RWTS' (Read or Write Track and Sector) routine. When you initialize a disk, DOS calls the 'RWTS' which writes the address and data marks for each track and sector. If you change the address or data marks in the 'RWTS' core then any disk you INIT from this core will have the changed address and data marks and will be uncopyable, at least by normal methods (such as the COPYA program on your System Master disk).

The particular locations for DOS 3.3 are 47413 and 47262. These are only two of a number of locations. The normal values at these addresses are 222. When you change this byte, be sure to change both locations to the same value. Use a value between 160 and 250.

Follow this routine:

1. BOOT NORMAL DOS
2. LOAD YOUR 'HELLO' PROGRAM
3. INSERT A BLANK DISK IN YOUR DRIVE
4. ISSUE THE COMMANDS:
   POKE 47413, (NEW NO.)
   POKE 47262, (NEW NO.)
5. INIT (NAME OF YOUR HELLO PROGRAM)

You now have a protected disk! The normal Apple copy programs will not work on this diskette. To use it, you must boot this disk; this will load the modified DOS.

To load programs onto this disk, you must change the two new locations that you POKE'd earlier.

Follow these steps:

1. BOOT NORMAL DOS
2. LOAD YOUR PROGRAM
3. POKE 47413, X
4. POKE 47262, X
5. INSERT THE PROTECTED DISK IN THE DRIVE
6. SAVE (PROGRAM NAME)

where: X = the number used earlier in step 4, above.

Follow these six steps to load your programs to the new copy-protected disk.

After you have transferred all your programs to the protected disk, you need to consider one final point. In order to get the full benefit of this method of copy protection, a special 'HELLO' program should be used. This program should set the RUN flag as well as lock out the RESET switch and 'CTRL-C', and then it should RUN your program.

An example (insert these lines in your 'HELLO' program

```
0  ONERR GOTO 1
1  POKE 214, 120:
   POKE 1010, 102:
   POKE 1011, 213:
   POKE 1012, 112:
```

Lines 0 and 1 set the RUN command, and lock out the 'CTRL-C' and RESET keys.

I trust these techniques are of some interest. Be certain to practice them with disks which do not contain valuable data!!

A NEW DOS?

the DOS Mouse

The rumors about a new DOS (Disk Operating System) for the Apple have finally been confirmed by Apple. The July 18 issue of InfoWorld carried a story announcing Apple's new ProDOS. While ProDOS will not be released until early 1984, Apple has already licensed it to software developers. This writer has already acquired a copy of the new ProDOS and can shed some light on these developments.

First, ProDOS will be able to move files to and from the Apple][/][+/][e and the Apple ///. Apparently, ProDOS programs will not be interchangeable between the][and the ///, but the data files can be swapped. In addition, ProDOS uses a hierarchical file structure (like the ///, unlike the][) which will support multiple sub-directories on a single disk. With this structure you can refer to disks by **name**, and no longer need to append a messy 'D2' to refer to a disk in the second drive; Apple Pascal, by the way, also uses a similar structure.

ProDOS will also support 'interrupts' from programs and external peripherals. Interrupts let external processes 'grab' the computer's attention to request specific applications, such as a request to stop processing numbers and start reading data coming over a local area network (**hint hint!**). I suspect ProDOS will also be compatible with Apple's Profile hard disk since Profile communicates to the computer through interrupt requests (another hint hint).

The big question on everybody's mind is whether ProDOS requires additional hardware. The answer is an emphatic no **provided** your Apple has at least 64K and the DOS 3.3 ROM chips supplied on the disk interface card.

This DOS Mouse is impressed with the new operating system. Unfortunately, this DOS Mouse does not have the ProDOS reference manuals, and is still trying to discover all the new features and commands offered by this system. You can count on seeing additional write-ups as the situation develops!

SOFTWARE AND HARDWARE FOR APPLE COMPUTERS AT DISCOUNT PRICES.

HARDWARE:
*RAM CARD *Z80 SOFTCARD
*MONITORS
*VERBATIM,3M,SYNCOM,MAXELL DISKETTES
*EPSON PRINTERS*OKIDATA PRINTERS
*DIABLO PRINTERS*NEC SPINWRITERS

SOFTWARE:
*SILICON VALLEY SYSTEMS*MUSE
*PENGUINE(GRAPHICS SYSTEM)
*MICROSOFT*VISICORP(VISICALC)
***AND MANY OTHERS**

SOME OF OUR PRICES:
EPSON FX-80 $540 RX-80 $365
BOX OF VERBATIM DISKETTES $25

KAPADIA COMPUTER MART
9,LAREDO DRIVE
CHELMSFORD,MA.01824.
TEL:(617)-256-1533

REVIEW: TG TRACKBALL

BY THE WINERS
SHARON, STUART AND BETTE

We wanted to give a favorable review of the TG Trackball, we really did, but we found the controller almost impossible to use with any of the more popular games. We cannot recommend it to anyone for any purpose.

The problem with the trackball is that it is designed to work as a joystick, but doesn't have the joystick's most important characteristics. There is a significant reduction in sensitivity relative to the TG Joystick; it took a movement of over six inches to have the same effect on the games that a two inch movement of the TG Joystick would have produced. However, the most serious problems were that there was NO maximum excursion of the trackball from the center and it is impossible to tell, roughly, how far you are from some reference point.

As with the joystick, when you roll the ball to the right, the value of PDL(0) (or PDL(1) depending on how you are holding the unit) increases up to a maximum of 255. It is this value that the program uses to control the play of the game. There is a maximum movement of the joystick that roughly corresponds to this maximum value of PDL(0); there is no corresponding maximum to the position of the trackball. You can keep on rolling the trackball to the right indefinitely. When you want to reverse direction, you must roll the trackball all the way back to the point when PDL(0) BEGINS to equal 255 before the game will respond. That means that if you have overshot 255 by 10 rolls of the ball, you must return 10 rolls of the ball before there can be any response. This is unacceptable in any of the games that we tried to play with the trackball. The aliens (or whatever) destroy you while you are trying to get the trackball to respond.

The second serious problem was that you could never tell how far you had moved the ball from some reference point. In those games that require that you compensate for some external field for momentum, it proved impossible to calibrate the ball so that you could eventually ignore it. This problem could be partially solved by painting a dot onto the ball for a reference before you began to play. But this tends to defeat the whole idea behind the trackball.

Another, relatively minor problem was that the position of the two buttons on the TG Trackball were reversed relative to their positions on the TG Joystick. We probably would have gotten accustomed to it eventually, but we fail to understand why TG had to make things so difficult.

JUST THE BASICS

By Jack Sheehan

Recently we have experienced a change of leadership at NEAT Our new president, Terry Scheller and officers have been asked by our former officers to take over the direction of our club. Along with new blood there may be some new ideas.

To help assist in this change I will be writing this column. I presented two in-service courses to the teachers in the Waltham school system last year. The name of the course was COMPUTER LITERACY-AN INTRODUCTION TO MICROCOMPUTERS. Many of the ideas used were a result of the information obtained from our monthly meeting our newsletter, our special inter est groups, and our disk of the month.

That is what this column is about - an introduction to the APPLE microcomputer. The column is geared to the computer novice and/or the person new to micro's. This author's hope is that it will impart a computer awareness and help you to use your APPLE as an effective tool.

To begin with let's examine the initial package most users purchase. The most common package consists of the APPLE machine a disk drive, 48K or 64K of memory, and a video screen. This package represents the five physical devices that make up a computer. There are five logical ideas behind any computer. They are an input idea or unit, an out put unit, a memory unit, an arith metical and logical unit, and a central processing unit.

INPUT UNIT-allows the user to com municate with the mach ine and makes that information available to the other units so it (the information) is processed. Usually the keyboard or disk drive is the device that supplies input.

OUTPUT UNIT-takes the information that has been processed and displays it on an output device such as a videoscreen or a printer.

MEMORY UNIT-the concept of memory is most troublesome for beginners. Suffice it to say this unit retains information and makes it available to the other processing units when it is needed.

ARITHMETIC AND LOGICAL UNIT (ALU) this unit is responsible for performing calculations and mak ing logical decision s. On the APPLE it is housed inside the 6502 microprocessor.

CENTRAL PROCESSING UNIT (CPU) this unit is the com puter's coordinator and is responsible for supervising the operation of the oth er units. It acts like the quarterback of the computer.

The rest of this column is easy. There are only a few more terms we need to deal with to develop a beginning vocabulary.

A COMPUTER PROGRAM is a set of instructions to the computer in a particular sequence that performs a task. Therefore, a game is a

computer program. A word processor is a computer program and the PHONE LIST on the DOS 3.3 is a computer program.

These programs are referred to as SOFTWARE. The HARDWARE are the "hard" items you purchased-the machine, the disk drive, and the monitor. A computer system consists of hardware, software, and you. The most important component is YOU. Afterall, the computer didn't buy you!

Let's close this month's column by discussing some fundamental commands.These commands should help unravel some of the mystery of your computer. To execute them just type them as they appear and press RETURN.

CATALOG-displays the contents of a disk.

TEXT - used after running a graphics program. Restores the video screen to the text mode from the graphics mode.

HOME - clears the screen and places the cursor in the upper left hand corner of the screen.

CTRL C- in most programs will allow you to break away or exit a program. N.B. TO EXECUTE A CONTROL COMMAND HOLD DOWN THE CTRL KEY AND TYPE THE LETTER AT THE SAME TIME.

CONT - will allow the program you just broke away from to resume executing.

CTRL S- will cause the program that is executing to pause. To resume the execution hit any key.

CTRL X-will print a reverse slash (\). Any data you were entering will be ignored by the computer and the reversed slash will be printed after it.The cursor will appear on the following line.

RUN f - f=program name. This will execute the named program

RUN - this will execute the program that is loaded in the computer's memory.

LIST - this will list the BASIC statements that make up the computer program.

LISTa,b-this will list the statements that begin at a and end at b.

Well, that does it for July! In the coming months I hope to help you unlock some of the mysteries of your APPLE. Before I sign off I would like to recommend a very useful text for beginners as well as some of our old salts. It is called the APPLE II USER'S GUIDE by Lon Poole. You should be able to obtain a copy at your friendly computer store.

NEW PRODUCT: ANALIST

Synoptic Software has introduced their ANA-LIST list processor software for the Apple. ANA-LIST's unique blend of data entry, editing, calculating, sorting, and reporting let you easily maintain lists and tables (such as vendor lists, mailing lists), analyze them from different points of view, and prepare custom reports for your particluar applications. (See their ad on the inside front cover.)

MACROSOFT: A REVIEW

by Dr. J.

Assembly language is probably the most cryptic language you can use on the Apple. The two major problems are that instructions in assembly language use non-descriptive nmemonic^, and the code is 'strung out'; take a look at my MACHINE LANGUAGE SPOKEN HERE column this month and see how my listing takes roughly 45 lines of assembly code (lines 102-146) to duplicate instructions which could be written using a single line in Applesoft. Frankly, I'm not at all surprised that most people have the good sense to stay away from assembly language.

Enter Micro-SPARC with a new product to assist the weary assembly language programmer: MACROSOFT. MACROSOFT is a library of special routines developed for Micro-SPARC's assembler (THE ASSEMBLER) which lets you write assembly code using a BASIC-like syntax. THE ASSEMBLER is a 'macro' assembler, meaning you can define special words (macros) which represent a particular sequence of programming steps; during assembly these macros are actually replaced in the code by the blocks of code they represent. MACROSOFT is a collection of macros written for THE ASSEMBLER which use nice nmemonic^, such as GOTO, GOSUB, PRINT, FOR-NEXT, etc., to represent instructions in assembly language.

FEATURES:

MACROSOFT supports both integer (two-byte), floating point (6-byte), and string expressions. For example, the instructions to add 5 to the integer variable 'X' and assign it to the integer variable 'Y' can be written as:

```
LET Y=X+5
```

Were these variables floating point instead of integer, the instruction would simply be:

```
FP Y=X+5
```

During assembly, both of these statements would be translated into a (long) list of assembly language commands.

MACROSOFT supports many of the same commands you find in Applesoft. MACROSOFT has a full array of 'input' statements (including INPUT, GET, KEYPRESS, PEEK), 'output' statements (including PRINT, PR, POKE, SOUND), screen control statements (HOME, TEXT, FLASH, INVERSE, NORMAL, HTAB, VTAB) and special math functions (including DIV2, MULT2, ABS, RND, SGN). MACROSOFT also supports all the normal LORES and HIRES commnd^, plus many control statements, some of which are not even found in Applesoft (including GOTO, GOSUB, FOR-NEXT, IF-THEN-ELSE, WHILE-WHILE^, REPEAT-UNTIL).

With MACROSOFT, THE ASSEMBLER produces fast, albeit long, code. I ran a benchmark writing a bubblesort routine in Applesoft, MACROSOFT, and straight assembly language. The MACROSOFT version was sixteen times faster than the Applesoft

version, and only 20% slower than the straight assembly code.

Shortcomings? MACROSOFT has several shortcomings for me as an experienced assembly language programmer. I do feel some of these shortcomings will not be noticed by the novice - intermediate programmer. First, you cannot pass Applesoft integer and floating point variables directly to MACROSOFT. Instead, all variables used by a MACROSOFT assembly program must be stored in their own separate area in memory, and are not readily available to external Applesoft programs. Second, MACROSOFT does not support floating point arrays; don't expect to do some fancy scientific number crunching through MACROSOFT. Third, MACROSOFT does not let you use string arrays; don't expect to convert a text editor written in BASIC over to assembly language. Finally, MACROSOFT only supports one-dimensional (integer) arrays. Again, when I write assembly code, I do it to gain a significant speed advantage over Applesoft, and this advantage comes primarily when I do operations on 2-D arrays.

On the plus side, MACROSOFT is indeed easy to use, and can be used to write disk utilities, games (those which would otherwise be written in BASIC), and fast graphics programs. I have been using MACROSOFT for several months now, and feel it has cut my programming time by a factor of two or more for small 'utility' programs. By the way, Dave Szetela from MicroSPARC told me that a new version of MACROSOFT will soon be out which can pass variables to and from Applesoft.

MACHINE LANGUAGE SPOKEN HERE

By Dr. J

The doctor is in.

Well, as promised, this month we are going to see how to perform some advanced calculations on Applesoft arrays from assembly language. This month, we will develop a machine language program to add two two-dimensional arrays and store the results in a third array. That is, we are going to simulate the following Applesoft program in machine language:

```
10 FOR J = 0 TO N
20 FOR I = 0 TO M
30 C(I,J) = A(I,J) + B(I,J)
40 NEXT I
50 NEXT J
```

Our program will be able to add any two arbitrary arrays, each having M+1 rows and N+1 columns. [Remember, when we dimension an array as: A(M,N) in Applesoft, the array will really have M+1 rows and N+1 columns.]

A nice thing about array addition (or array subtraction, for that matter) is that we can use **linear addressing** to address individual elements in an array. That is, with the addition algorithm we can start at the very first element in the very first column in both array A( ) and array B( ), add them together, and store them in the very first element in the very first column in array C( ). We then continue the addition for each of the elements in the first column. When we have completed the first column, we can move to the very first element in the second column and continue the addition. When we

are finished with the second column we will move to the third column, and so on. Linear addressing, then, can be used any time the operation on the array moves sequentially from one element to the next without any interruptions.

In the main body of our machine language program, we will begin by picking up the pointers to the elements A(0,0), B(0,0), and C(0,0). We will add the element in A(0,0) to the element in B(0,0) and store the result in C(0,0). The next elements in the sequence [A(1,0), B(1,0), and C(1,0)] are each stored in memory 5 bytes away from the first elements; all we need to do is add '5' to the first pointers and we will obtain the pointers to the second elements. If we store these pointers, we can later add another 5 byte 'offset' to obtain the pointers to the third elements, and so on.

One last note before we actually examine the assembly listing. We will use the same special initialization routines we introduced last month to allow us to call the matrix addition routine by name from Applesoft BASIC. [This routine will not make much sense if you missed last month's column. If this is the case, drop me a line and I'll try to send you a copy of last month's column.]

The assembly listing is shown in Listing 1; the Applesoft program which calls this routine is shown in Listing 2. We begin the machine code program (lines 34-43) by setting the Ampersand (&) vector to point to a location in the program called 'START'. This location (lines 44-56) finds the address to the main matrix addition routine (MATADD), converts the value into a floating point number (line 47) and transfers the value of this number to the floating point variable passed in the Applesoft statement in listing 2:

```
& MADD
```

After this Applesoft statement has been executed, the value of the variable MADD will equal the starting address of the main matrix addition routine.

The main routine begins at line 60. The routine first picks up the pointers to the three arrays and two variables passed to it by the Applesoft statement:

```
CALL MADD,A(0,0),B(0,0),
   C(0,0),M,N
```

Lines 60-74 picks up the pointers to the three arrays. Lines 79-84 find the pointers to the variable M, move its value into FAC, convert it to a one-byte integer, and add 1 to it (to convert from M columns to M+1 columns). Lines 90-95 repeat the same steps for the variable N.

Finally, lines 102-135 actually handle the actual matrix addition. Lines 106-114 add the 'current' elements in A( ) and B( ) and store the result in the current element in C( ). Lines 115-129 add the 5-byte offset to the pointers to the current element to obtain the pointers to the next element, and store these new pointers. Subroutine ADDFIVE simply adds 5 to the 2-byte number stored in the ACCUMULATOR (low byte) and the Y-register (the high byte).

Benchmarks? I ran this program against an equivalent Applesoft

program and found the machine language version offers a speed advantage of 16:1!! If you frequently do matrix additions, you will certainly want this little routine in your library!

Well, that about wraps it up for now. Next month we will examine matrix multiplication.
Till then, good luck!

The doctor is out.

LISTING #2

```
10  D$ = CHR$(4)
15  REM RUN THE MACHINE LANG.
    PROGRAM
20  PRINT D$"BRUN MATRIX ADD.
    OBJ0"
30  N=30:M=30
40  DIM A(M,N),B(M,N),C(M,N)
50  FOR I=1 TO M:FOR J=1 TO N
    A(I,J)=1:B(I,J)=1:NEXT J,I
55  REM SET THE VARIABLE MADD
60  & MADD
65  REM CALL THE MATRIX ADD
    ROUTINE
70  CALL MADD,A(0,0),B(0,0),
    C(0,0),M,N
```

```
SOURCE FILE: MATRIX ADD
                      1 ;*****************************
0000:                 5 ; THIS ROUTINE WILL ALLOW US
0000:                 6 ; TO ASSIGN SPECIFIC NAMES TO
0000:                 7 ; OUR MACHINE LANGUAGE ROUTINES,
0000:                 8 ; AND THEN CALL THESE ROUTINES
0000:                 9 ; BY NAME FROM APPLESOFT.
0000:                10 ;
0000:                11 ; THE INSTRUCTIONS ARE PROVIDED
0000:                12 ; IN THE ACCOMPANYING ARTICLE
0000:                13 ;
0000:                14 ;*****************************
0000:                15 ;
0000:                16 ; SET UP POINTERS
0000:                17 ;
DFE3:                18 PTRGET  EQU  $DFE3     ; GET VARIABLE POINTER
DEBE:                19 CHKCOM  EQU  $DEBE     ; CHECK FOR COMMA
EAF9:                20 MOVFM   EQU  $EAF9     ; MOVE (Y,A) INTO FAC
EB2B:                21 MOVMF   EQU  $EB2B     ; MOVE FAC INTO (Y,X)
E2F2:                22 GIVAYF  EQU  $E2F2     ; FLOAT INTEGER IN A,Y
E6FB:                23 CONINT  EQU  $E6FB     ; CONVERT FAC TO INTEGER; STORE IN X-REG
E7BE:                24 FADD    EQU  $E7BE     ; ADD FAC TO THE NUMBER
0000:                25 ; POINTED TO BY (Y,A) AND KEEP IN FAC
0000:                26 ;
0000:                27 ; BEGIN THE PROGRAM
0000:                28 ;
----- NEXT OBJECT FILE NAME IS MATRIX ADD.OBJ0
9400:                29         ORG  $9400
9400:                30 ;
9400:                31 ; FIRST, SET UP THE AMPERSAND
9400:                32 ; '&' POINTER BY LOADING A 'JMP' INSTRUCTION
9400:                33 ; AT MEMORY LOCATION $3F5.
9400:A9 10           34         LDA  #)START   ; GET THE LOW BYTE OF
9402:8D F6 03        35         STA  $3F6      ; THE STARTING ADDRESS
9405:A9 94           36         LDA  #(START   ; GET THE HIGH BYTE
9407:8D F7 03        37         STA  $3F7
940A:A9 4C           38         LDA  #$4C      ; OP-CODE FOR 'JMP'
940C:8D F5 03        39         STA  $3F5      ; STORE IT
940F:                40 ;
940F:                41 ; FINISHED, RETURN TO BASIC
940F:                42 ;
940F:60              43         RTS            ; RETURN TO BASIC
```

PASCAL: DISK NOTES

by Jim Wrenholt

(Reprinted from The Scarlett Letter, published by the Big Apple User's Group.)

I've been spending my spare time lately learning my way around the Pascal system on my Apple. After reading through an introductory book on the subject, I thought myself ready to study some actual Pascal programs. Well, after five minutes of fiddling around I succeeded only in obliterating my disk's directory. I still don't know what magical set of keystrokes causes the disaster.

Being a veteran of such DOS catastrophes, my first thought was to get out my Disk ZAP and fix the disk. Pascal lives on the disk in 16 sector format, so that my ZAP (track/sector read/write routine) is capable of reading it. After spending a little time looking, I became convinced that my program's textfiles were still intact and that only the directory was messed up.

Now to rebuild the directory. First, I needed to discover the layout of the directory. Pascal is not DOS and the Pascal reference manuals are not as complete as the DOS manual either. So scrounging through my old magazines, I turned up an article in CALL APPLE from March/April 1981 that showed how the directory is formatted on the disk.

All that is needed is a name and the beginning and ending block number. Pascal numbers are stored in consecutive blocks so

```
9410:A9 94      44 START  LDA  #<MATADD  ; GET THE HIGH BYTE TO
                45 ; THE 'MATIX ADD' ROUTINE
9412:           46        LDY  #>MATADD  ; GET THE LOW BYTE
9412:A0 1F      47        JSR  GIVAYF    ; CONVERT THE INTEGER IN
9414:20 F2 E2   48 ; A,Y AND STORE IN FAC
9417:           49        JSR  PTRGET    ; GET POINTER TO FIRST
9417:20 E3 DF   50 ; PARAM. IN THE AMPERSAND STATEMENT
941A:           51        TAX            ; ACC -> X-REG.
941A:AA         52        JSR  MOVMF     ; MOVE FAC TO (Y,X)
941B:20 2B EB   53 ;
941E:           54 ; FINISHED, RETURN TO BASIC
941E:           55 ;
941E:60         56        RTS            ; RETURN TO BASIC
941F:           57 ;
941F:           58 ; MATIX ADD ROUTINE
941F:           59 ;
941F:20 BE DE   60 MATADD JSR  CHKCOM    ; REMOVE FIRST COMMA
9422:20 E3 DF   61        JSR  PTRGET
9425:8D CF 94   62        STA  ALOW      ; STORE LOW BYTE POINTER
9428:8C D0 94   63        STY  AHIGH     ; STORE HIGH BYTE POINTER
942B:           64 ;
942B:           65 ; NOW REPEAT FOR THE MATICES 'B' & 'C'
942B:           66 ;
942B:20 BE DE   67        JSR  CHKCOM
942E:20 E3 DF   68        JSR  PTRGET
9431:8D D1 94   69        STA  BLOW
9434:8C D2 94   70        STY  BHIGH
9437:20 BE DE   71        JSR  CHKCOM
943A:20 E3 DF   72        JSR  PTRGET
943D:8D D3 94   73        STA  CLOW
9440:8C D4 94   74        STY  CHIGH
9443:           75 ;
9443:           76 ; NOW READ THE FOURTH ARGUMENT, 'M',
9443:           77 ; AND DETERMINE THE NUMBER OF ROWS
9443:           78 ;
9443:20 BE DE   79        JSR  CHKCOM
9446:20 E3 DF   80        JSR  PTRGET
9449:20 F9 EA   81        JSR  MOVFM     ; MOVE THIS VALUE INTO FAC
944C:20 F8 E6   82        JSR  CONINT    ; X-REG = INT(FAC)
944F:E8         83        INX            ; X-REG = X-REG + 1
9450:8E D7 94   84        STX  M         ; THE BYTE 'M' NOW CONTAINS THE
9453:           85 ; NUMBER OF ROWS PLUS ONE
```

there is no track/sector list to locate. But since Disk ZAP deals in tracks and sectors the real challenge became to discover how the block numbers are mapped onto the track sectors. The results of my sleuthing are shown in the following table:

BLOCK no.	FIRST trk/sec	SECOND trk/sec
x0	y/0	y/E
x1	y/D	y/C
x2	y/B	y/A
x3	y/9	y/8
x4	y/7	y/6
x5	y/5	y/4
x6	y/3	y/2
x7	y/1	y/F
x8	z/0	z/E
x9	z/D	z/C
xA	z/B	z/A
xB	z/9	z/8
xC	z/7	z/6
xD	z/5	z/4
xE	z/3	z/2
xF	z/1	z/F

Where:

y = 2 * x ; z = 2 * x + 1

It turns out that this block number to track/sector mapping is rather twisted. Remember that it takes two sectors for one block.

To convert from block number to track/sector you must think in hex, or at least convert. For instance, to find block number $53 on the disk, find X3 in the first column. the X is $5 in our case. Now double X to get Y, Y = 2 * X = 2 * $5 = $A. Finally, the next two columns give the track/sector pair Y/9 and Y/8. In our example Y = $A, so sectors 9 and 8 on track $A

```
                        ;
                        ; FINALLY, READ THE FIFTH ARGUMENT, 'N',
                        ; AND DETERMINE THE NUMBER OF COLUMNS
                        ;
9453:20 BE DE           JSR CHKCOM
9456:20 E3 DF           JSR PTRGET
9459:20 F9 EA           JSR MOVFM
945C:20 FB E6           JSR CONINT
945F:E8                 INX
9460:8E D8 94           STX N       ; THE BYTE 'N' NOW CONTAINS THE
                        ; NUMBER OF COLUMNS PLUS ONE
                        ;
                        ; WE HAVE READ ALL THE ARGUMENTS IN
                        ; THE LIST; NOW LET'S GET DOWN TO
                        ; SOME SERIOUS MATIX ADDITION
                        ;
9463:AD D8 94           LDA N
9466:8D D6 94           STA J
9469:AD D7 94 FORJ      LDA M
946C:8D D5 94           STA I
946F:AD CF 94 FORI      LDA ALOW
9472:AC D0 94           LDY AHIGH
9475:20 F9 EA           JSR MOVFM   ; FAC = A(I,J)
9478:AD D1 94           LDA BLOW
947B:AC D2 94           LDY BHIGH
947E:20 BE E7           JSR FADD    ; FAC = A(I,J)+B(I,J)
9481:AE D3 94           LDX CLOW
9484:AC D4 94           LDY CHIGH
9487:20 2B EB           JSR MOVMF   ; C(I,J) = FAC
948A:AD D0 94           LDA AHIGH
948D:AC CF 94           LDY ALOW
9490:20 C5 94           JSR ADDFIVE
9493:8D D0 94           STA AHIGH
9496:8C CF 94           STY ALOW
9499:AD D2 94           LDA BHIGH
949C:AC D1 94           LDY BLOW
949F:20 C5 94           JSR ADDFIVE
94A2:8D D2 94           STA BHIGH
```

are the desired locations.

As another example, let's locate block $3C. First, locate XC = $3C in the first column. Using the formula Z = 2 * X + 1 = 7, the block is found on sectors 7 and 6 of track 7.

To go the other way you must decide if the track is even or odd. If the track is even, use the top half of the chart; if odd, use the bottom half.

For an example of this, let's find the block corresponding to track/sector $0C/$0F. First $0C is an even number, so use the top half of the table. Next, X = $0C/2 = $6. Locating Y/F in the top half and moving over to the first column shows that track sector $0C/$0F is the second half of block number $67.

There is just one more thing to be remembered. When you find the start of your actual text on the disk you need to back up two blocks to leave room for the header which is at the start of all text files. For more information see the Apple Pascal Reference Manual 3.2.1.: The Format of Text Files.

Now to rebuild the directory. First make a backup of your Pascal disk. I use COPYA from the DOS System Master for this. (Of course, if I would have backed up in the very beginning I would not have had to go through all of this, but that's hindsight.) Any direct modification of sectors is considered dangerous so we need to back up now for sure.

The Pascal disk directory is divided into entries that are 26 (decimal) bytes long. The first

```
94A5:8C D1 94  124          STY BLOW
94A8:AD D4 94  125          LDA CHIGH
94AB:AC D3 94  126          LDY CLOW
94AE:20 C5 94  127          JSR ADDFIVE
94B1:8D D4 94  128          STA CHIGH
94B4:8C D3 94  129          STY CLOW
94B7:CE D5 94  130          DEC I
94BA:F0 03     131          BEQ SETJ
94BC:4C 6F 94  132          JMP FORI
94BF:CE D6 94  133 SETJ     DEC J
94C2:D0 A5     134          BNE FORJ
94C4:60        135          RTS                 ; RETURN TO BASIC
94C5:          136 ;
94C5:          137 ; ADDFIVE ROUTINE
94C5:          138 ;
94C5:48        139 ADDFIVE PHA                 ; SAVE HIGH BYTE
94C6:98        140          TYA                ; LOAD LOW BYTE INTO ACC
94C7:18        141          CLC                ; CLEAR CARRY
94C8:69 05     142          ADC #5             ; ADD FIVE BYTES
94CA:A8        143          TAY                ; STORE IN Y-REG
94CB:68        144          PLA                ; RECOVER HIGH BYTE
94CC:69 00     145          ADC #0             ; COMPLETE ADDITION
94CE:60        146          RTS
94CF:          147 ;
94CF:          148 ; DEFINE LOCAL STORAGE
94CF:          149 ;
94CF:00        150 ALOW    DFB 00
94D0:00        151 AHIGH   DFB 00
94D1:00        152 BLOW    DFB 00
94D2:00        153 BHIGH   DFB 00
94D3:00        154 CLOW    DFB 00
94D4:00        155 CHIGH   DFB 00
94D5:00        156 I       DFB 00
94D6:00        157 J       DFB 00
94D7:00        158 M       DFB 00
94D8:00        159 N       DFB 00
```

26 bytes are for the disk volume name. Each subsequent group of 26 bytes is for an individual file. The directory starts on track $0, secto. $B and extends downward. There is room for up to 77 file names.

THE FIRST 26 BYTES

BYTE	%
1,0	START OF DIRECTORY
3,2	END OF DIRECTORY BLOCK NUMBER
5,4	ALWAYS ZERO
6	LENGTH OF DISK VOL. NAME
7-13	DISK VOLUME NAME STRING
15,14	NUMBER OF BLOCKS ON DISK
17,16	ENTRIES IN THE DIRECTORY
19,18	ALWAYS ZERO
21,20	CURRENT DATE
23,22	ALWAYS ZERO
25,26	ALWAYS ZERO

INDIVIDUAL FILES

BYTE	%
1,0	BLOCK START NUMBER
3,2	BLOCK END NUMBER
5,4	FILE TYPE: 2=CODE, 3=TEXT, 5=DATA
6	LENGTH OF FILE NAME
7-21	FILENAME STRING
23,22	NUMBER OF BYTES IN THE LAST BLOCK
25,24	FILE DATE

As an example, I have included a sector dump of my Apple I diskette. Notice, as usual, the low order byte is first. Also, the ending block number is really the beginning of the next file or one block past the actual end of the text file.

REAR GUARD

mini'app'les

A review

by Hugh Kurtzman

Rear Guard is a hi res game in which you must shoot down alien ships to survive. There are 6 different aliens that give you from 25 to 300 points when you destroy them. There are also two Satellites. One with 'fuel' the other with 'shields'. By zaping one of these you will add to your fuel or shield supply. You may play the game with paddles, joysticks, or the keyboard. The ESC key freezes the action. Your score is saved and inserted into the list by rank.

I am a beginner at games so I can't compare this game with others. I found the one page instruction sheet enough to get started and was able to 'increase my score almost every time I played. I tried the game with paddles and the keyboard and found I could get higher scores with the paddles. I don't own a joystick so couldn't try that mode. Several young people I asked to try Rear Guard said it was 'ok', but it didn't challenge them very long!

I did not figure out how to trade pods for fuel, shields and/or bonus points. There is a pointer that is used to accomplish this.

Rear Guard requires one disk drive, 48k of memory, and DOS 3.3.

Adventure International has a good replacement and backup policy. They will replace any disk that will not boot free the first year and for $5.00 after the first year. They also included a coupon that could be used to purchase a backup for $3.99.

COMPUTERS etc.

216 NEWBURY STREET (RT. 1 SOUTH), PEABODY, MA 01960

(617) 535-5252

"Your Computer Support Headquarters"

Franklin *1776* Special

Includes:

- Franklin Ace 1000 with 64K (runs all Apple II software)
- 80 column card
- Disk drive with controller
- 12" green screen BMC video display
- Built in fan
- Built in numeric keypad
- ACE WRITER word processing program
- ACE CALC electronic spreadsheet program
- 100 useful and entertaining software programs
- 10 blank disks
- 1 disk storage container and index file
- Choice of 2 game programs OR reconditioned computer table
- 3 hours of training with our Educational Consultant (two 1.5 hour sessions)

ALL FOR ONLY $1776.00

Computer Desk Special --- $79.95

We also carry APPLE, COLUMBIA (IBM compatible), & SEIKO computers as well as APPLE, EPSON, & GEMINI printers.

Whether you buy a disk or a system, COMPUTERS ETC. intends to provide you with the support you need.

FRANKLIN
COMPUTER CORPORATION

MEMBERSHIP APPLICATION

1. Name _____ Date _____
2. Street _____
3. City _____ State _____ Zip _____
4. Phones (a) Home _____ (b) Work _____
5. Software background _____
6. Hardware background _____
7. What will you use your APPLE for? _____
8. In what area can the club assist you? _____
9. Are you available to assist other members? _____
10. Total amount enclosed _____

1983 Membership Dues are

UNITED STATES..... $ 15.00

CANADA..... $ 21.00

(OFFICE USE ONLY)

PAID BY: cash check RECEIPT #_____ STATUS: New Renewal MEM #_____

-DISKETTE ORDER FORM-

Diskettes are sold to members and non-members and can be picked up at the meeting or mailed anywhere in the continental U.S. Circle desired selection.

Games	I	II	III	IV	V	VI	VII					
Education	I	II	Math		I	Scientific			I	Applesoft Tutor		
Pascal	I	II	III	IV	V	VI						
D.O.M.1980					M	J	J	A	S	O	N	D
D.O.M.1981	J	F	M	A	M	J	J	A	S	O	N	D
D.O.M.1982	J	F	M	A	M	J	J	A	S	O	N	D

All D.O.M.'s up to the current month $5.00 picked up, $6.50 mailed. The current D.O.M. is copied free on your initialized diskette at the meeting.

All catagorized diskettes include a description of the programs and are available for $5.00 if picked up and for $6.50 if they must be mailed.

Non-members prices are $10.00 picked up and $11.50 mailed.

Check here if you want diskettes mailed ( ).

NAME _____

ADDRESS _____

CITY/STATE _____

ZIP _____

MEMBERSHIP CARD # _____

Make check payable to N.E.A.T.
Send to:
N.E.A.T.-Attn: Librarian
P.O.Box 2652
Woburn, MA 01801

Total enclosed $ _____.____

neat
New England Apple Tree
Box 2652, Woburn, MA 01801

BULK RATE
U.S. POSTAGE
PAID
NEWTONVILLE, MA. 02160
PERMIT NO. 57931

EXPIRES 02/84

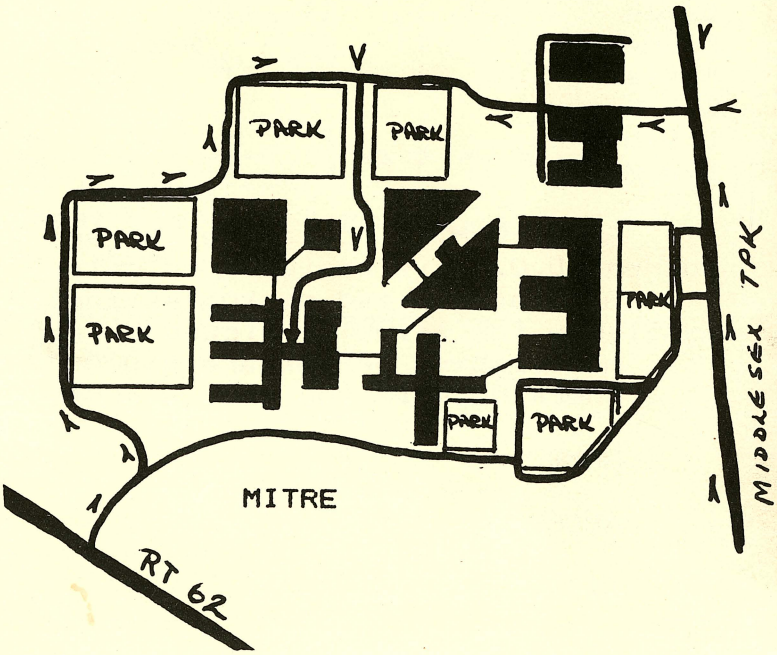
ADVERTISING RATES

Single Issue
- Full Page.....$40.00
- Half Page.....$20.00
- Quarter Page.....$10.00

3 Month series or Longer
- Full Page.....$30.00
- Half Page.....$15.00
- Quarter Page.....$7.50

Inserts letter size accepted at the same rates, subject to weight limitations.

Personal advertisements of members, e.g., swap notices, personal articles for sale, etc. accepted without charge.

Please send ad copy in black and white (half-tones are permited but no bleeds) by the last Wednesday of the month preceeding the month of issue to:

N.E.A.T. Notes
Attention: Editor
P.O.Box 2652
Woburn, MA. 01801